

# Inpainting at Modern Camera Resolution by Guided PatchMatch with Auto-Curation

Lingzhi Zhang<sup>1</sup>, Connelly Barnes<sup>2</sup>, Kevin Wampler<sup>2</sup>, Sohrab Amirghodsi<sup>2</sup>,  
Eli Shechtman<sup>2</sup>, Zhe Lin<sup>2</sup>, and Jianbo Shi<sup>1</sup>

<sup>1</sup> University of Pennsylvania

<sup>2</sup> Adobe Research

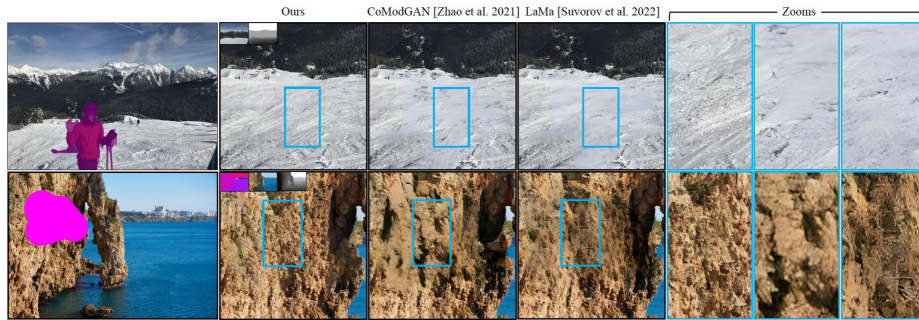


Fig. 1: Inpainting at modern camera resolutions via guided PatchMatch (guides shown in inset) with a novel automatic curation module. Photos are 12 and 30 MP. Our result has *significantly* higher fidelity high-res detail than CoModGAN [45] and LaMa [27], the strongest baselines according to our user study, which were upsampled by Real-ESRGAN [31]. The guides on the top left of the second columns indicate the chosen guides for the specific image.

**Abstract.** Recently, deep models have established SOTA performance for low-resolution image inpainting, but they lack fidelity at resolutions associated with modern cameras such as 4K or more, and for large holes. We contribute an inpainting benchmark dataset of photos at 4K and above representative of modern sensors. We demonstrate a novel framework that combines deep learning and traditional methods. We use an existing deep inpainting model LaMa [27] to fill the hole plausibly, establish three guide images consisting of structure, segmentation, depth, and apply a multiply-guided PatchMatch [1] to produce eight candidate upsampled inpainted images. Next, we feed all candidate inpaintings through a novel curation module that chooses a good inpainting by column summation on an 8x8 antisymmetric pairwise preference matrix. Our framework’s results are overwhelmingly preferred by users over 8 strong baselines, with improvements of quantitative metrics up to 7.4 *times* over the best baseline LaMa, and our technique when paired with 4 different SOTA inpainting backbones improves each such that ours is overwhelmingly preferred by users over a strong super-res baseline.

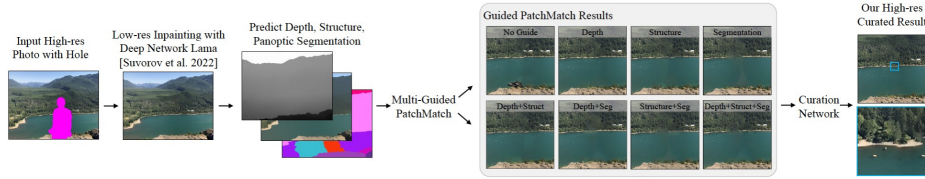


Fig. 2: Overview of our framework. See the intro for discussion of components.

## 1 Introduction

Image inpainting involves removing a region and replacing it with new pixels so the modified photo is visually plausible. We develop a method that can take an off-the-shelf low-res inpainting deep model and extend it to modern camera resolutions. Applied to LaMa [27], this yields a new a SOTA method for inpainting at modern camera resolutions that dramatically outperforms all existing models.

Traditional patch-based synthesis approaches, such as Wexler et al. [33], Barnes et al. [1], and Darabi et al. [6] were used for high-quality image inpainting at arbitrary resolutions. Recently, the state-of-the-art for *low resolution* image inpainting has been advanced by deep convolutional methods, such as Zeng et al. [40], Zhao et al. [45], and LaMa [27]. When we compared these methods, we noticed an interesting trade-off. Patch-based methods such as Patch-Match [1] can synthesize high-quality texture at arbitrarily high resolution but often make mistakes regarding structure and semantics. On the other hand, deep convolutional methods tend to generate inpaintings with plausible structure and semantics but less realistic textures. Worse still, most deep methods are limited to output resolutions such as 512 or 1024 pixels image along the max dimension<sup>3</sup>, *much smaller* than modern camera resolutions which are typically 4K or above.

Based on our observations, we first collected a dataset of 1045 high-quality photos at 4K resolution or above representative of modern sensors and paired them with freeform and object-shaped hole masks. This dataset has approximately *2 orders of magnitude* more pixels per image than the dataset Places2 [46] that is commonly used for deep inpainting evaluation. Interestingly, photos in the Places2 dataset often used for evaluation have fewer pixels than the world’s first digital camera [5], the FUJIX DS-1P, produced in 1988, so common evaluation practices are **more than 3 decades** behind sensor technology. We encourage the community to evaluate on our dataset, since datasets such as Places2 [46] are *not representative* of modern camera sensors, and which may therefore mislead researchers by giving improper guidance to the community.

After establishing the dataset, we next married the complementary advantages of patch-based synthesis in generating high texture quality at arbitrarily high resolution and deep networks, which can better predict structure and semantics. Our hybrid pipeline works as illustrated in Figure 2. We first inpaint the hole region using an off-the-shelf deep network, such as LaMa [27], at 512

<sup>3</sup> With the exception of HiFill [36] and LaMa [27], which we discuss in related work.

pixels along the long edge of the image. This method typically establishes a reasonable semantic layout and structure inside the hole. Still, as can be seen in Figure 1, the texture quality is often poor if we zoom in. Next, we extract several guide images using existing methods: depth [37], structure [35], and panoptic segmentation [17]. We then use these guides in a multi-guided PatchMatch [1] implementation to perform patch-based image inpainting. Depending on the input photo, the best result may be obtained by different combinations of guides, so we produce a set of eight guided PatchMatch *candidate inpainting* results using multiple combinations of guides.

Besides the novelty of our overall framework, our key technical innovation is creating a **novel automatic curation module** whose architecture is shown in Figure 3(a). This curation module automatically selects one good inpainted image out of the 8 candidates by an architecture suited for making subtle comparisons and contrasts between similar images. It does this by constructing an antisymmetric 8x8 matrix whose entries are populated in pairs by a network that learns to predict *antisymmetric pairwise human preferences* for every possible pair of candidate inpaintings. The preferred inpainted image is selected by taking the row with max column sum for the 8x8 matrix. Different from previous work such as RealismCNN [51] and LPIPS [43] and image quality assessment papers, our network estimates pairs of entries  $M_{ij}$  and  $M_{ji}$  at a time for an 8x8 matrix  $M$  that is *antisymmetric* i.e.  $M = -M^T$ , and the pairwise structure is critical because it allows the network to differentiate between subtle feature differences. On pairs of images, our curation has near-human performance.

We conducted quantitative experiments with metrics commonly used to evaluate deep networks and multiple user studies for our evaluation. Our method dramatically outperforms state-of-the-art deep inpainting methods according to our quantitative metrics and user studies. User preferences are between 79% and 96% for our method, *every one of our 7* user study results in Tables 2-4 has statistically significant preference for ours, and quantitative metrics improve by up to 7.4 times over strong baselines like LaMa [27] coupled with Real-ESRGAN [31]. Because our method uses patch-based synthesis, it can easily *scale to arbitrary resolution*: in the supplemental, we show results on images up to 62 megapixels. We show in our experiments that our method can be combined with four different deep inpainting baselines and improves **every** one of them for modern camera resolution inpainting according to quantitative metrics and user studies. Most convolutional models cannot scale to modern camera resolutions, except HiFill [36] and LaMa [27], which our evaluation shows we outperform.

Our paper contributes: (1) A benchmark dataset of 4K+ resolution images with holes appropriate for evaluating inpainting methods as they perform on modern camera sensors and evaluation of 9 methods on this dataset; (2) A novel high-resolution inpainting framework which shows that deep inpainting models need not give high quality results only on low resolutions. This framework has a choice of guides that perform well, which were nontrivial to choose and required extensive empirical investigations as discussed in Section 4.2, and is the first to explore combinations of multiple guides with a mechanism to automatically

choose between them; (3) A curation module (seen for the first time in inpainting) with near-human performance that chooses a good inpainted image in a new way by populating an 8x8 antisymmetric preference matrix by comparing images one pair at a time and column-summing that matrix; (4) State-of-the-art results on the natural image inpainting task from both quantitative and user studies, with dramatic improvements over 8 strong baselines; (5) Our method can be combined with a variety of SOTA deep inpainting baselines and improves each of them for modern camera resolution inpainting. To enable reproducibility, we plan to release our full benchmark dataset including results of all methods and curation module test code and weights.

## 2 Related Work

**Patch-based synthesis and inpainting.** Our approach uses the patch-based inpainting method of Wexler et al. [33] as implemented in the PatchMatch framework [1]. Inspired by image analogies [12], we added to this basic inpainting framework multiple guiding channels. Similar guided texture synthesis approaches have been used for stylizing renderings [9] and stylizing video by example [2,15], but the choice of guides is *nontrivial and application-dependent*: we performed many empirical investigations of alternative options to finally choose the guides used in this paper, as we discuss later in Section 4.2. Image Melding [6] improved patch-based synthesis results with geometric and photometric transformations and other means. We incorporated a gain and bias term from that paper to obtain better inpaintings for regions with smooth gradient changes in intensity. Kaspar et al. [16] performed texture synthesis using a patch-based optimization with guidance channels that preserve large-scale structures. Several papers explored initialization and search space constraints that use scene-level information that is different from ours [11,13,7].

Recently, some papers have explored learning good features by making PatchMatch differentiable [8,47]. Because of our modern camera resolution photos, multi-res and many iterations of filling used in PatchMatch [1], and because differentiable PatchMatch techniques track multiple candidate patches, the GPU memory requirements of applying a differentiable PatchMatch naively in our context are far beyond the capacity of today’s GPUs. Thus, we use a non-differentiable PatchMatch with off-the-shelf guide features, and rely on the downstream curation module to pick good guide features.

**Neural network image inpainting.** One significant advantage of convolutional neural network inpainting methods is that they can gain some understanding of semantic information such as global and local context, edges, and regions. Context encoders used a CNN to fill in a hole [24] by mapping an image with a square hole to a filled image. Iizuka et al. [14] used two discriminators to encourage the global and local appearance to both be reasonable. Yu et al. [38] proposed a contextual attention model which can effectively copy patches from outside the hole to inside the hole, however, it is limited in resolution because of the brute-force dense attention mechanism. Liu et al. [19] and Yu et al. [39]

improved inpainting results by masking and gating convolutions, respectively. Zhao et al. [45] better address the situation of large holes by co-modulating the generative decoder using both a conditional and stochastic style representation.

A number of papers recently attempt to separate the scene in terms of edges [22,34], structure-texture separation, such as Ren et al. [26] which uses relative total variation (RTV) [35], or Liu et al. [20]. Our approach is inspired by these works and uses structure from RTV [35] and a panoptic segmentation map to guide PatchMatch.

A few recent works attempt to train neural networks that can output higher resolution results. ProFill [40] uses a guided upsampling network at a resolution of up to 1K on the long edge. HiFill [36] introduced a contextual residual aggregation module that weights residuals similar to those in a Laplacian pyramid [4], at up to 8K resolution, but according to our evaluation its results are worse than our method. The work LaMa [27] was trained on 256x256 patches but can be evaluated on images up to 2K.

**Visual realism.** For pretraining our curation network, we use a similar idea as RealismCNN [51], which learns a visual realism score for composite images using a large dataset of images and a synthetic compositing pipeline. In our case, however, we generate millions of fake inpainted images using our pipeline, and pretrain a network to classify these images as fake and real images as real. Inspired by LPIPS [43] and image quality assessment (IQA) papers (e.g., [3,28,49]), we then fine-tune on real human preferences between pairs of synthetic inpainted images, but we use a different architecture and inference that involves predicting entries one pair at a time in an *antisymmetric matrix* and column-summing it. In contrast, LPIPS [43] learns a *symmetric* distance metric, full-reference IQA papers typically are also estimating some perceptual distance that is not anti-symmetric, and no-reference IQA papers use a single image as input. See Figure 3 for an illustration. Also related is Wang et al. [30], which trains a “universal” detector that can distinguish between CNN-generated images and real images.

### 3 Inpainting Dataset

To benchmark the performance of our method and 8 other methods, we collect 1045 high-quality images at modern sensor resolutions from two sources: DIV8K [10], and a test set portion of the dataset of photos taken by the authors and their collaborators mentioned earlier in Section 4.3. The photos span a diverse variety of scenes including indoors and outdoor urban photography including many architectural styles, nature and wildlife, macro photos, and human subject photos. The photos are all at least 4K pixels along the long edge. The mean megapixel count is 20 and the maximum is 45, excluding high-res panos that go up to 62 MP. DIV8K [10] has previously been used for super-resolution tasks and contains images with resolutions up to 8K. We chose all 583 images in DIV8K that were above 4K resolution on the long axis.

For each test image, with equal probability, we sample either a free-form mask or an instance mask using the same process and hole dataset as in ProFill

[40]. Different from ProFill, because our photos are from modern sensors, we generate larger holes with a mean hole size of 2.3 MP: see the supplemental for details. No test images are seen during training. To enable reproducibility, we plan to release the testing dataset and results for all methods. Experimental results are shown later in Section 5.2.

## 4 Our Hybrid Synthesis Method

### 4.1 Multi-guided PatchMatch for Image Inpainting

PatchMatch [1] is an efficient randomized correspondence algorithm that is commonly used for patch-based synthesis for images, videos [2,15] and neural feature maps [18,48]. For image synthesis, PatchMatch repeatedly performs matching from the region being synthesized to a reference region: in our case, the matching is from the hole to the background region. We implement the method of Wexler et al. [33], with default hyperparameters for PatchMatch (e.g. 7x7 patches) and Wexler et al. One key advantage of PatchMatch is that it can efficiently scale to arbitrary resolutions while preserving high-texture fidelity. We extend this basic method in two ways: we allow multiple guides to be used (as in [12]), and we implement the gain and bias term from Image Melding [6].

For the multiple guides, we modify the sum of squared differences (SSD) used in Equation (5) of PatchMatch. Instead of computing SSD over a 3 channel color image, we compute a weighted SSD over a  $3 + m$  channel image with channel weight  $w_i$ , where the first 3 channels are RGB and remaining channels are guides.

$$w_i = \begin{cases} w_c/3 & i \leq 3 \\ (1 - w_c)/m & i > 3 \end{cases} \quad (1)$$

Using a separate validation set of images and manual inspection, we tried different settings of  $w_c$  and empirically found the best setting is  $w_c = 0.6$  if there is no structure guide, otherwise  $w_c = 0.3$ . Structure and RGB information are highly correlated so we decrease the RGB weight if there is a structure guide.

We implement the gain and bias term from Image Melding [6] because we find it helps improve inpainting quality in cases where there are subtle gradients, such as in the sky. We implement this term using RGB color space.

### 4.2 Guides Used for PatchMatch

We performed extensive investigations of many possible guides, and empirically settled on the three selected in this paper — structure, depth, and panoptic segmentation — because they worked the best on our validation set. Although our main novel low-level technical contribution is our curation module that we describe in the next section, the existence of appropriate guides, the empirical work needed to choose them and our overall novel multi-guided inpainting framework with curation also forms a high-level technical contribution.



Our pipeline works by first inpainting the hole using an off-the-shelf deep network LaMa [27] at 512 pixels on the long edge and then extracting three guide images: depth, segmentation, and structure. Depth is a useful cue as regions at a similar distance from the camera have usually more relevant content. This is especially important for slanted surfaces (e.g. the ground) where scale and focus properties of texture vary with depth. We also often want to sample semantically similar content, motivating the segmentation guide. However, the segmentation labeling might not be fine-grained enough to distinguish between different types of floor tiles or wall colors. Therefore we add the structure guide that captures the main edges and color regions in the image, and abstracts away the texture.

For depth prediction, we used a recent method by Yin et al. [37], which is retrained using a DPT [25] backbone to obtain better results. For panoptic segmentation prediction, we used PanopticFCN [17] retrained with a ResNeSt [41] backbone, which obtains a higher panoptic quality of 47.6 on the COCO val set. For structure prediction, we extract the structure using RTV [35], that was shown effective for inpainting by Ren et al. [26].

We performed extensive investigations of many alternative possible guides, and choose these three because they gave the best quality. We tried raw RGB color instead of structure, but this produced worse results due to patch matching becoming less flexible (see supplemental for details), Iterative Least Squares texture smoothing [21], but the latter had color bleeding artifacts that caused worse inpaintings, and Gaussian blurred RGB color and bilateral filtered RGB instead of structure, but these produced artifacts in the patch synthesis. We also tried different segmentation and depth modules and settled on the above ones because they gave better patch synthesis results.

Is there a universal ‘best’ guide for all images? We find that different choices of guides may give the best final inpainting result, depending on the input image and hole. Empirically, we find that the structure guide can help PatchMatch find patches with consistent edges and fine-scale structures, depth guides can be handy when the images have gradually changing depth such as outdoor photography, and segmentation guides can be useful when the segmentation map is accurate for preventing patches from leaking into the wrong semantic or object region. In general, we find no single guidance map by itself is sufficient to get the best quality of results, and multiple guidance maps are needed. In many applications it is desirable to have a fully automatic image inpainting process and we found that a fixed weighted combination of guides leads to sub-optimal results. Thus, as shown in Figure 2, we generate a variety of results using different guides and use our curation module to automatically choose a good one. Specifically, we use a simple scheme where we can either enable or disable each of our 3 guides, so the total number of possibilities for guides are  $2^3 = 8$ , including the use of no guides, and then we select among those eight generated results. We show an example of how the guides influence the results in the supplemental.

### 4.3 Curation Module

Suppose you are given two inpainted candidate images and asked, “Which of these do you prefer, or do you have no preference?” We were inspired by how humans carry out this task: since the images usually look similar, we carefully compare and contrast between subtle differences in visual features to determine which image is slightly better overall. Our architecture thus is designed to enable such *subtle comparisons and contrasts* between features within inpainted images. In Figure 3, we show an architecture diagram of our novel curation module and a comparison with other common architectures such as LPIPS [43] and no-reference IQA. Notably, our curation module has a different architecture and inference methodology that populates entries in a matrix  $M$  that is *antisymmetric* due to the paired preference task and column-sums that matrix to determine the relative preference vector of one inpainting candidate as contrasted with others.

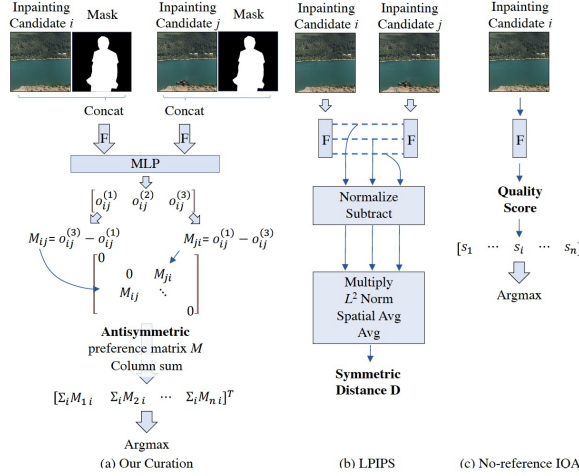


Fig. 3: Our curation architecture produces scores in an **antisymmetric** matrix  $M$  that is column-summed. LPIPS [43] produces a **symmetric** distance score. Architectures used in RealismCNN [51] and no-reference IQA produce a score for each candidate image that is **independent** of the other candidates.

**Pretraining.** Inspired by RealismCNN [51], our curation network backbone  $F$  is first pretrained to classify for a given image, whether it is a real image or a fake inpainted image as output by our pipeline. Our reasoning is that we observed that the initial pretrained network predictions have correlation with human perception of inpainting quality, and can allow the network to learn good features over a very large number of photos, but we need to later fine-tune on human preferences to obtain performance close to humans. We generated a dataset for pretraining the curation network by collecting 48229 diverse pho-



tos that are at least 2K in resolution, generating 10 synthetic holes for each, and then generating 8 guided PatchMatch results. This results in more than 3 million inpainted images at 2K resolution. Our curation network backbone  $F$  is EfficientNet-B0 [29]. We modify the input to take 4 input channels and pretrain the network from scratch. We trained for one epoch using a binary cross-entropy loss, after which we obtained train, test, val accuracies of 98.9%, 99.3%, 99.2%, respectively. Please see the supplemental for more details.

**Fine-tuning on human preferences.** In nearly all cases, the pretrained curation network  $F$  can easily distinguish between real and our inpainted results, but it was not specifically trained to predict human preference among different inpainted results. Therefore, we next fine-tune our network for a paired preference task. By subsampling the dataset described in the last subsection, and comparing sampled pairs of the 8 guided PatchMatch results against the others, we gather approximately 33000 synthetic inpainted image pairs for which we gather human preferences. We discuss in our supplemental lessons learned in gathering these preferences. For each image pair, our model works by featurizing each of the images in the pair through a shared-weight pretrained EfficientNet backbone  $F$ , and then using a small MLP to predict 3 classes that the human preference data contain: prefer left image, tie, prefer right image.

In contrast to perceptual distances such as LPIPS [43], we have a different task where our model predicts an antisymmetric preference. In particular, if one swaps the left and right image, one would expect the preference for left or right image to also swap. We thus impose this swapping as a data augmentation, by doubling each original batch to include a swapped copy of the batch: we found this accelerates and stabilizes training. We include a variety of standard augmentations that we list in the supplemental.

**Inference for curation.** Our network is trained on paired preferences, but at inference time, we want to compare 8 guided PatchMatch inpaintings and establish a preference for each, and a preferred ordering. Moreover, in alternative implementations, one might wish to compare more or fewer images. Thus, to compare  $n$  inpainted candidate images, we form an  $n \times n$  matrix  $M$ , where  $M_{ij}$  is the probability of preferring method  $i$  over method  $j$ . We establish this probability for all pairs  $i, j$  with  $i < j$  by setting  $o_{ij}^{(k)}$  for  $k = 1, 2, 3$  as the 3 softmaxed outputs of the MLP for the pair, and then compute  $M_{ij} = o_{ij}^{(3)} - o_{ij}^{(1)}$  and  $M_{ji} = -M_{ij}$ . The ground truth and prediction are antisymmetric matrices i.e.  $M = -M^T$ . The preference of a given inpainted image  $i$  *in the context of the other images* is the average of row  $i$  of  $M$ . In this way, we recover the same antisymmetric paired preferences in the special case of  $n = 2$ , but also generalize to establishing preferences among arbitrary numbers of images.

The input resolution for our curation network is 512x512, however, the photographs to be inpainted can have 1 to 2 orders of magnitude more pixels. We use an operation called “auto crop” to resize a crop region around the hole that contains approximately 25% hole pixels and surrounding context to the target resolution. Please see the supplemental for details about automatic cropping.

## 5 Experiments

### 5.1 Curation Module

Method	Accuracy	Accuracy for Easy Cases
Human Performance	57.1% (+0.7%)	86.1% (+1.4%)
<b>Our Curation Network</b>	<b>56.4%</b>	<b>84.7%</b>
Ours No Pretraining	53.9% (−2.5%)	78.8% (−5.9%)
Ours Fewer Augmentations	53.8% (−2.7%)	77.8% (−6.9%)
Ours No Mask	52.6% (−3.8%)	72.2% (−12.5%)
Ours Late Fusion Variant	52.6% (−3.9%)	82.1% (−2.6%)
Ours Early Fusion	51.4% (−5.1%)	77.8% (−6.9%)
Ours Freeze Backbone	43.7% (−12.7%)	53.0% (−33.1%)
NIMA[28] w/ MLP	41.4% (−15.0%)	50.0% (−34.7%)
MetaIQA+[50] w/ MLP	41.1% (−15.3%)	50.4% (−34.3%)
Random Chance	33.1% (−23.3%)	30.6% (−54.2%)

Table 1: Curation network performance on human paired preference data, ablations, and comparisons. We report two test set accuracies: over the whole dataset, and over only easy cases. Our network is competitive with humans and outperforms all alternatives. Parenthetical numbers are relative to our network.

We show in Table 1 the performance on human paired preference data for our curation module, ablations, and comparisons. The table is computed from the human preference dataset previously discussed in Section 4.3. In our user study, if a human expresses a preference for one image, we ask if the preference is strong or weak. Because the task itself is challenging, we also report **easy cases** as those where mean human preference is strongly for one image.

We list our network and human performance in Table 1. Human preference is determined by collecting an additional opinion for a random subset of images. Our curation module outperforms all other alternatives and is only slightly worse than human performance by both metrics. Although the accuracy for both humans and ours is “only” 56-57%, this is already much better than random chance at 33%, and this is because of the difficulty of the task, where for many fills it is hard to tell whether they are tied in quality or one or the other is preferred. For easy or unambiguous cases, the accuracies for humans and ours are both much better, at 85-86%. Since models can overfit to the training set, we always report the checkpoint with highest validation accuracy.

We next discuss the ablations in Table 1. “Ours No Pretraining” skips the pretraining, which is necessary for best generalization. “Ours Fewer Augmentations” is an ablation where removing JPEG compression, rotation, and noise reduces accuracy. For “Ours No Mask,” we do not input the hole mask. “Ours Late Fusion Variant” modifies the pretraining so instead of using one classifier, both real and fake image are featurized with a shared-weight EfficientNet backbone and compared with an additional MLP. “Ours Early Fusion” modifies the

network by concatenating both images with mask and feeding this through a single EfficientNet backbone followed by MLP.

We find that fine-tuning only the MLP for the human preference task and freezing the backbone network weights is insufficient. “Ours Freeze Backbone” freezes backbone weights after pretraining and only fine-tunes the MLP. Similarly, “NIMA[28] w/ MLP” and “MetaIQA+[50] w/ MLP” use pretrained, frozen no-reference image quality assessment backbones, and fine-tune the MLP.

Our Table 4 shows that the guided inpainting chosen by our curation module out of all 8 options outperforms both a random choice of a guided fill and Photoshop’s Content-Aware Fill: the outperformance in user preference is particularly strong. That table is for the inpainting dataset, which is described next.

## 5.2 Comparison with Other Methods

Methods	LPIPS ↓	FID ↓		P-IDS ↑		U-IDS ↑		User Pref. ↑	
		Full	Patch	Full	Patch	Full	Patch	Full Image	Boundary Patch
EdgeConnect [22]	0.05017	35.06	41.05	0.04	4.56	0.00	0.55	-	-
Deepfillv2 [39]	0.05295	32.87	36.06	5.54	5.47	1.35	0.84	-	-
MEDFE [20]	0.05170	33.97	60.87	0.48	2.23	0.00	0.26	-	-
HiFill [36]	0.05213	34.39	31.74	4.15	5.20	0.75	0.97	-	-
CoModGAN [45]	0.05099	24.81	32.08	14.72	7.01	4.47	1.51	28	17
MADF [52]	0.04773	23.62	33.21	10.48	6.81	2.14	1.48	6	12
ProFill [40]	0.04783	24.25	31.26	11.35	6.89	2.26	1.31	10	16
LaMa [27]	0.04588	19.20	35.95	17.24	6.86	5.62	1.38	28	22
SuperCAF (Ours)	0.04156	18.74	15.63	22.46	19.77	10.70	10.22	128	133

Table 2: A comparison study with the state-of-the-art inpainting methods. The top 3 methods are colored: 1, 2, 3.

We compare our algorithm with eight state-of-the-art image inpainting methods, quantitatively and qualitatively. Among all these methods, HiFill [36] can run on images up to 8K resolution like our method, the work LaMa [27] states that they can generalize to higher resolutions up to around 2K, ProFill [40] can run on images up to 1K resolution, and the rest of the methods can only run on images up to 512 x 512.

The main focus of our method is inpainting of high resolution images of size 4K and beyond. We thus generate full resolution images for all methods. We attempted to make the comparison as generous as possible for baseline methods by applying Real-ESRGAN [31] for super-resolution to increase all methods with limited output resolution back to the native image resolution. We chose Real-ESRGAN [31], since it is the state-of-the-art SR algorithm for real-world images and is robust to visual artifacts in the input. We apply HiFill [36] at native resolution, and ProFill [40] at its maximum of 1K resolution. For LaMa [27], we found that although it can be applied at higher resolutions, we obtain best quality by applying it at a resolution where the maximum axis is 512 pixels.

Methods	LPIPS ↓	FID ↓	User Pref. ↑
ProFill [40] + SR [31]	0.0478	24.2589	21
ProFill [40] + Ours	<b>0.0425</b>	<b>20.4362</b>	<b>179</b>
CoModGAN [45] + SR [31]	0.0510	24.8189	36
CoModGAN [45] + Ours	<b>0.0430</b>	<b>19.9224</b>	<b>164</b>
MADF [52] + SR [31]	0.0477	23.6290	27
MADF [52] + Ours	<b>0.0421</b>	<b>19.9022</b>	<b>173</b>
LaMa [27] + SR [31]	0.04588	19.2022	42
LaMa [27] + Ours	<b>0.04156</b>	<b>18.7414</b>	<b>158</b>

Table 3: A pairwise comparison study between high-resolution inpainting results for upsampled by Real-ESRGAN [31] and results upsampled by our framework, for four recent inpainting methods. Users are asked to choose the best image from a pair for each inpainting model. The best score is bold.

Methods	LPIPS ↓	FID ↓	User Pref. ↑
Content-Aware Fill	0.04675	23.0068	16
Random Guided PM	0.04281	19.7704	24
SuperCAF (Ours)	<b>0.04156</b>	<b>18.7414</b>	<b>60</b>

Table 4: A comparison study with Photoshop’s Content-Aware Fill, which is based on PatchMatch [1] and a randomly selected Guided PatchMatch baseline. Photos average 20 megapixels and holes average 2.3 megapixels. The user studies are performed on 100 photographs where all method outputs are different from each other. The best scores are bold text in the table.

Additionally, we tried two scenarios for the baselines: running on the auto-crop region discussed earlier in Section 4.3 and running on the full image. In our evaluation we found no advantage for the baselines run on the auto-crop region so we used the full image scenario.

For the quantitative evaluation, we evaluated six popular metrics: Peak Signal-to-Noise Ratio (PSNR), SSIM [32], LPIPS [44], the recently improved version of FID [23], and P-IDS and U-IDS from CoModGAN [45], which were recently shown to correlate highly with human perception. Because the holes are very large (2.3 megapixel on average) and valid fills can have quite diverse contents that differ greatly from the original image, we feel the metrics *FID and P/U-IDS are most appropriate*, and we also show LPIPS, and we report PSNR and SSIM in the supplemental. The quantitative results are shown in Table 2. For P/U-IDS, because of the dataset size of 1045 images, we also apply vertical and horizontal augmentation for the full image to improve convergence of those metrics.

In Table 2, we report quantitative metrics for two scenarios: “full” indicates a square crop region around the entire inpainted region was used, as determined by auto-crop Section 4.3, and “patch” indicates 10 smaller randomly sampled 256x256 crop regions drawn at consistent position from locations where the patch center is a hole pixel. We note that ours outperforms all baselines by the metrics in the table. We particularly note that in the patch scenario, for the metrics FID, P-IDS, and U-IDS, we *dramatically* outperform the SOTA method LaMa [27] by factors *between 2.3 to 7.4 times*, and the outperformance can be even greater for other methods. This is because our method has much higher texture fidelity at the finest resolutions, since it can copy relevant background patches via PatchMatch. These textures form a coherent whole, as indicated by outperformance via other metrics.

As often mentioned in the inpainting literature (e.g., [38,39]), image inpainting lacks a good quantitative evaluation metric, and there is no single metric that can be used to gauge real image quality. Thus, we also conduct two user studies for randomly sampled 200 test images in comparison with the top four methods using Amazon Mechanical Turk. For the first user study, we use the “full” image scenario described earlier, and in the second user study, we use a “boundary patch” setup. The boundary patch is determined by randomly sampling a 512x512 crop region, centered on a boundary pixel of the hole region. This setup allows human participants to easily contrast the texture synthesized inside the hole with that in the background region, and thus assess the suitability of methods for inpainting at modern camera resolutions. In the user study, we give user output images from all methods with randomized order, and ask users to pick the best image for each case. For each of the two user studies, we recruited more than 150 users and asked each user to evaluate a randomly sampled batch of 20 images from the whole test set. The study results are shown in the last two columns of Table 2. In the “full” scenario, our automatically selected guided PatchMatch outperforms the other methods by a large margin of *4.6 times to 21 times*, and in the “boundary patch” scenario, our method outperforms alternatives even more strongly, by *6 times to 11 times*. Users in general very strongly prefer our method for inpainting at modern camera resolutions, but in the crop scenario where a user focuses on high-resolution detail — as might be important for large displays or large format prints — our method performs better still.

In Table 3, we show our method can be used in combination with 4 different baseline methods that perform the initial coarse-scale inpainting: ProFill[40], CoModGAN [45], MADF [52], and LaMa [27]. In every case, our method outperforms the baseline with Real-ESRGAN [31] super-resolution applied. We ran user study containing pairs of images, and found the user study preferences strongly prefer our method, with between *4.5 and 8.5 times* higher preference for ours over the baseline. Our method is suitable when combined with a variety of deep inpainting baselines, and greatly improves user preference over alternatives. We show similar results for 3 older inpainting models in the supplemental.

In Table 4, we compare our method to two other baselines: Photoshop’s Content-Aware Fill (CAF), which is based on PatchMatch [1], and a baseline that randomly picks with equal probability one of our eight guided PatchMatch results. Our method again performs best for the quantitative metrics. We ran a user study on 100 images where all methods have different outputs, and again find our method is *strongly preferred 3.4 to 4 times* more than the other two baselines. This indicates that our method outperforms a strong commercial baseline of Photoshop’s CAF, which is used by professionals to manipulate photos at modern camera resolutions, and shows that our curation outperforms a random guided PatchMatch result. We include many photographic results in our supplemental material, and show that the preference for ours is statistically significant for *all 7* of the above user studies.

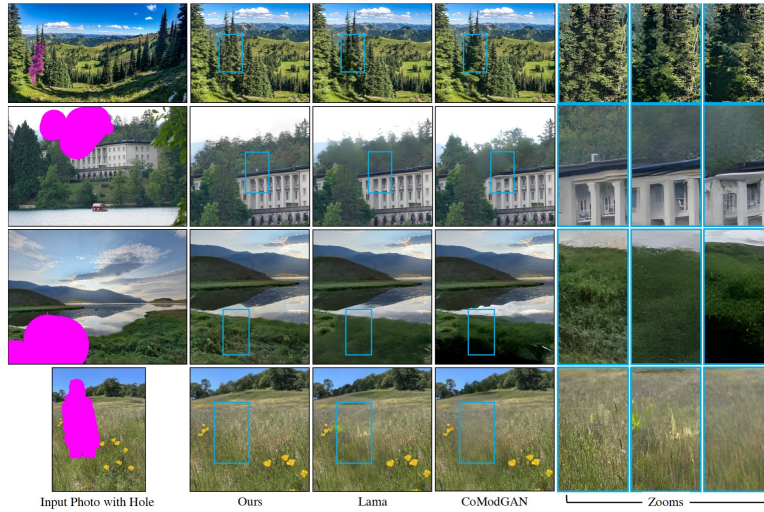


Fig. 4: Results for our method and two baselines with Real-ESRGAN, for (from top): a 26 MP nature panorama with real hole, a 20 MP photo with random scribble hole, a 12 MP lake photo with random scribble hole, a 12 MP field photo with an instance hole. Please check out supplemental for visual results.

### 5.3 Running Time

On a 3.6 GHz 8 core Intel i9-9900K with 11 GB NVidia RTX 2080 Ti, for a representative 12 MP image with 4K resolution, our naive implementation takes 23.0 seconds, and our optimized implementation takes 2.5 seconds by initially computing PatchMatch results at 1K, running curation, then using another PatchMatch to obtain the 4K result.

## 6 Discussion, Limitations, Future Work

Our method has some limitations, which could be mitigated through user interactions such as manually picking guides. Generally, PatchMatch is good at synthesizing texture and repetitive regular structures, but structures under perspective transformations in architecture can be broken. We occasionally observe small amounts of blur especially near the hole boundary: this might be addressed by using curation in a smarter way such as an iterative fill [40,42]. Occasionally, we observe repetitions of salient patches: these might be mitigated by incorporating patch usage budgets [9,16] combined with saliency. GANs may produce amazing results by hallucinating content not present in the input image, but they can also hallucinate bizarre artifacts. We use patch-based synthesis throughout the image, however, patch-based synthesis can remove unique features, so the result could be allowed to deviate from patch-based synthesis if we believe a hallucinated output is a good one.



## References

1. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* **28**(3), 24 (2009) [1](#), [2](#), [3](#), [4](#), [6](#), [12](#), [13](#)
2. Bénard, P., Cole, F., Kass, M., Mordatch, I., Hegarty, J., Senn, M.S., Fleischer, K., Pesare, D., Breeden, K.: Stylizing animation by example. *ACM Transactions on Graphics (TOG)* **32**(4), 1–12 (2013) [4](#), [6](#)
3. Bosse, S., Maniry, D., Müller, K.R., Wiegand, T., Samek, W.: Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Transactions on image processing* **27**(1), 206–219 (2017) [5](#)
4. Burt, P.J., Adelson, E.H.: The laplacian pyramid as a compact image code. In: *Readings in computer vision*, pp. 671–679. Elsevier (1987) [5](#)
5. Cade, D.: The world’s first ‘fully’ digital camera was created by fuji (2016), <https://petapixel.com/2016/06/09/photo-history-worlds-first-fully-digital-camera-invented-fuji/> [2](#)
6. Darabi, S., Shechtman, E., Barnes, C., Goldman, D.B., Sen, P.: Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on graphics (TOG)* **31**(4), 1–10 (2012) [2](#), [4](#), [6](#)
7. Diamanti, O., Barnes, C., Paris, S., Shechtman, E., Sorkine-Hornung, O.: Synthesis of complex image appearance from limited exemplars. *ACM Transactions on Graphics (TOG)* **34**(2), 1–14 (2015) [4](#)
8. Duggal, S., Wang, S., Ma, W.C., Hu, R., Urtasun, R.: Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 4384–4393 (2019) [4](#)
9. Fišer, J., Jamriška, O., Lukáč, M., Shechtman, E., Asente, P., Lu, J., Šykora, D.: Stylit: illumination-guided example-based stylization of 3d renderings. *ACM Transactions on Graphics (TOG)* **35**(4), 1–11 (2016) [4](#), [14](#)
10. Gu, S., Lugmayr, A., Danelljan, M., Fritsche, M., Lamour, J., Timofte, R.: Div8k: Diverse 8k resolution image dataset. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. pp. 3512–3516. IEEE (2019) [5](#)
11. He, K., Sun, J.: Statistics of patch offsets for image completion. In: *European conference on computer vision*. pp. 16–29. Springer (2012) [4](#)
12. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. pp. 327–340 (2001) [4](#), [6](#)
13. Huang, J.B., Kang, S.B., Ahuja, N., Kopf, J.: Image completion using planar structure guidance. *ACM Transactions on graphics (TOG)* **33**(4), 1–10 (2014) [4](#)
14. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)* **36**(4), 1–14 (2017) [4](#)
15. Jamriška, O., Sochorová, Š., Texler, O., Lukáč, M., Fišer, J., Lu, J., Shechtman, E., Šykora, D.: Stylizing video by example. *ACM Transactions on Graphics (TOG)* **38**(4), 1–11 (2019) [4](#), [6](#)
16. Kaspar, A., Neubert, B., Lischinski, D., Pauly, M., Kopf, J.: Self tuning texture optimization. In: *Computer Graphics Forum*. vol. 34, pp. 349–359. Wiley Online Library (2015) [4](#), [14](#)
17. Li, Y., Zhao, H., Qi, X., Wang, L., Li, Z., Sun, J., Jia, J.: Fully convolutional networks for panoptic segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 214–223 (2021) [3](#), [7](#)

18. Liao, J., Yao, Y., Yuan, L., Hua, G., Kang, S.B.: Visual attribute transfer through deep image analogy. *ACM Trans. Graph.* **36**(4), 120:1–120:15 (Jul 2017). <https://doi.org/10.1145/3072959.3073683>, <http://doi.acm.org/10.1145/3072959.3073683> 6
19. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 85–100 (2018) 4
20. Liu, H., Jiang, B., Song, Y., Huang, W., Yang, C.: Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. *arXiv preprint arXiv:2007.06929* (2020) 5, 11
21. Liu, W., Zhang, P., Huang, X., Yang, J., Shen, C., Reid, I.: Real-time image smoothing via iterative least squares. *ACM Transactions on Graphics (TOG)* **39**(3), 1–24 (2020) 7
22. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212* (2019) 5, 11
23. Parmar, G., Zhang, R., Zhu, J.Y.: On buggy resizing libraries and surprising subtleties in fid calculation. *arXiv preprint arXiv:2104.11222* (2021) 12
24. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 2536–2544 (2016) 4
25. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 12179–12188 (2021) 7
26. Ren, Y., Yu, X., Zhang, R., Li, T.H., Liu, S., Li, G.: Structureflow: Image inpainting via structure-aware appearance flow. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 181–190 (2019) 5, 7
27. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. *WACV: Winter Conference on Applications of Computer Vision* (2022) 1, 2, 3, 5, 7, 11, 12, 13
28. Talebi, H., Milanfar, P.: Nima: Neural image assessment. *IEEE Transactions on Image Processing* **27**(8), 3998–4011 (2018) 5, 10, 11
29. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: *International Conference on Machine Learning*. pp. 6105–6114. PMLR (2019) 9
30. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot... for now. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 8695–8704 (2020) 5
31. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 1905–1914 (2021) 1, 3, 11, 12, 13
32. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* **13**(4), 600–612 (2004) 12
33. Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. *IEEE Transactions on pattern analysis and machine intelligence* **29**(3), 463–476 (2007) 2, 4, 6

34. Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., Luo, J.: Foreground-aware image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5840–5848 (2019) [5](#)
35. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. *ACM transactions on graphics (TOG)* **31**(6), 1–10 (2012) [3](#), [5](#), [7](#)
36. Yi, Z., Tang, Q., Azizi, S., Jang, D., Xu, Z.: Contextual residual aggregation for ultra high-resolution image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7508–7517 (2020) [2](#), [3](#), [5](#), [11](#)
37. Yin, W., Zhang, J., Wang, O., Niklaus, S., Mai, L., Chen, S., Shen, C.: Learning to recover 3d scene shape from a single image. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR) (2021) [3](#), [7](#)
38. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5505–5514 (2018) [4](#), [13](#)
39. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4471–4480 (2019) [4](#), [11](#), [13](#)
40. Zeng, Y., Lin, Z., Yang, J., Zhang, J., Shechtman, E., Lu, H.: High-resolution image inpainting with iterative confidence feedback and guided upsampling. In: European Conference on Computer Vision. pp. 1–17. Springer (2020) [2](#), [5](#), [6](#), [11](#), [12](#), [13](#), [14](#)
41. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., et al.: Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955 (2020) [7](#)
42. Zhang, L., Zhou, Y., Barnes, C., Amirghodsi, S., Shechtman, E., Lin, Z., Shi, J.: Perceptual artifacts localization for inpainting. In: Proceedings of the European Conference on Computer Vision (ECCV) (October 2022) [14](#)
43. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018) [3](#), [5](#), [8](#), [9](#)
44. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018) [12](#)
45. Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E.I., Xu, Y.: Large scale image completion via co-modulated generative adversarial networks. In: International Conference on Learning Representations (ICLR) (2021) [1](#), [2](#), [5](#), [11](#), [12](#), [13](#)
46. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017) [2](#)
47. Zhou, X., Zhang, B., Zhang, T., Zhang, P., Bao, J., Chen, D., Zhang, Z., Wen, F.: Cocosnet v2: Full-resolution correspondence learning for image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11465–11475 (2021) [4](#)
48. Zhou, X., Zhang, B., Zhang, T., Zhang, P., Bao, J., Chen, D., Zhang, Z., Wen, F.: Full-resolution correspondence learning for image translation. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR)* (2021) [6](#)
49. Zhu, H., Li, L., Wu, J., Dong, W., Shi, G.: Metaiqa: Deep meta-learning for no-reference image quality assessment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14143–14152 (2020) [5](#)

50. Zhu, H., Li, L., Wu, J., Dong, W., Shi, G.: Generalizable no-reference image quality assessment via deep meta-learning. *IEEE Transactions on Circuits and Systems for Video Technology* (2021) [10](#), [11](#)
51. Zhu, J.Y., Krahenbuhl, P., Shechtman, E., Efros, A.A.: Learning a discriminative model for the perception of realism in composite images. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 3943–3951 (2015) [3](#), [5](#), [8](#)
52. Zhu, M., He, D., Li, X., Li, C., Li, F., Liu, X., Ding, E., Zhang, Z.: Image inpainting by end-to-end cascaded refinement with mask awareness. *IEEE Transactions on Image Processing* **30**, 4855–4866 (2021) [11](#), [12](#), [13](#)